

D. REMARKS**Status of the Claims**

Claims 1-26 are currently present in the Application, and claims 1, 10, 15, and 24-26 are independent claims. No claims have been amended, cancelled, or added in this Response.

Examiner Interview

Applicant wishes to thank Examiner Rose and Examiner Al-Heshami for the courtesy extended to Applicant's attorney during a telephone interview on April 20, 2006. During the interview, the Sayag reference (U.S. Patent No. 6,898,602) was discussed with regard to Applicant's claim 1. In particular, Applicant's attorney noted that Sayag does not teach "identifying a program statement in the program where the variable is last used," and then "inserting a nullification statement after the identified program statement, the nullification statement adapted to nullify the identified last-used variable." Rather, Sayag appears to execute a garbage collector ***immediately prior*** to a memory allocation. No agreement was reached on the claims.

Drawings

Applicant notes with appreciation the Examiner's acceptance of Applicant's drawings, filed with the Application on July 2, 2003.

Amendment to the Specification

The specification has been amended to correct an inadvertent, typographical error.

Claim Rejections - Alleged Anticipation Under 35 U.S.C. § 102

Claims 1-26 stand rejected under 35 U.S.C. § 102(e) as being anticipated by Sayag, U.S. Patent No. 6,898,602

(hereinafter Sayag). Applicant respectfully traverses the rejections under 35 U.S.C. § 102.

To anticipate a claim, the reference must teach every element of the claim (Manual of Patent Examining Procedure §2131). Applicant teaches and claims a method, system, and computer program product for automatically including a statement in program code that nullifies an object or variable. An activation record is compared with program statements to determine the last usage of such objects and variables. When an object or variable is last used in the program, a statement is automatically included in the program code nullifying the variable. Applicant respectfully submits that Sayag does not teach every element of Applicant's independent claims. Using claim 1 as an exemplary claim, claims 1, 10, and 15 include the following elements:

- reading one or more variables included in one or more activation records included in the computer program
- identifying a program statement in the program where the variable is last used
- inserting a nullification statement after the identified program statement, the nullification statement adapted to nullify the identified last-used variable

Note that independent claims 24-26 also include at least the elements listed above.

Sayag purports to enable a developer "to determine the amount of memory that is consumed by a running application at

any point of its execution" (col. 2, lines 18-21). Sayag executes a garbage collector immediately prior to a memory allocation (col. 2, lines 22-25). Using this approach means that "no memory allocation ever occurs until all unreachable memory has been reclaimed" (col. 2, lines 25-27). It is important to note that Sayag executes a garbage collector **immediately prior** to a memory allocation, which is in direct contrast to Applicant's claimed invention, where a nullification statement is inserted **after** a program statement in which a variable is last used.

In particular, Sayag does not teach "identifying a program statement in the program where the variable is last used," and then "inserting a nullification statement after the identified program statement, the nullification statement adapted to nullify the identified last-used variable," as taught and claimed by Applicant in independent claims 1, 10, 15, and 24-26. A close reading of Sayag does not reveal any discussion of determining where a variable is last used in a program. The Examiner cites Sayag at col. 3, lines 15-19 and col. 7, line 48 as teaching this element (see Office Action, page 3, lines 18-20). In col. 3, lines 15-19, Sayag discusses identifying locations in a program where garbage collection should be executed (see also, col. 3, lines 1-14). Sayag notes that one method for determining where to execute garbage collection is to designate locations in the code "that correspond to completion of consecutive statements of the computer program" (col. 3, lines 17-19). However, identifying consecutive statement in a computer program, in order to determine where to perform garbage collection, is not the same as identifying a program statement in a program **where a variable is last used**, as taught and claimed by Applicant.

In col. 7, line 48, Sayag discusses establishing a range of program instructions for testing. The range may include all instructions between the first and last instruction of the program, or it may be a smaller range (col. 7, lines 43-51). However, determining **a range of program instructions** for testing is not the same as identifying a program statement in a program **where a variable is last used**, as taught and claimed by Applicant. For the reasons set forth above, Applicant respectfully submits that Sayag does not teach "identifying a program statement in the program where the variable is last used," as taught and claimed by Applicant in independent claims 1, 10, 15, and 24-26.

Applicant further submits that Sayag does not teach "inserting a nullification statement after the identified program statement, the nullification statement adapted to nullify the identified last-used variable," as taught and claimed by Applicant in independent claims 1, 10, 15, and 24-26. Sayag does not discuss nullifying variables and certainly does not discuss a nullification statement that is inserted after an identified program statement in which a variable is last used. The Examiner cites Sayag at col. 5, lines 47-48, col. 7, lines 37-38, and col. 7, line 48 as teaching this element. Applicant respectfully disagrees.

In col. 5, lines 47-48, Sayag discusses embedding an application program in hardware or installing the application program in memory as software. In col. 7, lines 37-38, Sayag discusses installing an application in a development platform. As discussed above, in col. 7, line 48, Sayag discusses establishing a range of program instructions for testing. The range may include all instructions between the first and last

instruction of the program, or it may be a smaller range (col. 7, lines 43-51). Applicant is at a loss to understand how any of the cited portions of Sayag can be said to teach "inserting a nullification statement after the identified program statement, the nullification statement adapted to nullify the identified last-used variable," as taught and claimed by Applicant. It is important to note that Applicant does not claim simply inserting a nullification statement anywhere in a program. Applicant specifically claims identifying a program statement where a variable is last used, and then inserting a nullification statement after the identified program statement in order to nullify the identified last-used variable. Sayag simply does not teach or disclose these elements of Applicant's claims.

For the reasons set forth above, Applicant respectfully submits that independent claims 1, 10, 15, and 24-26 are not anticipated by Sayag. Therefore, Applicants respectfully requests that independent claims 1, 10, 15, and 24-26, and the claims which depend from them, be allowed.

Conclusion

As a result of the foregoing, it is asserted by Applicant that the remaining claims in the Application are in condition for allowance, and Applicant respectfully requests an early allowance of such claims.

Applicant respectfully request that the Examiner contact the Applicant's attorney listed below if the Examiner believes

that such a discussion would be helpful in resolving any remaining questions or issues related to this Application.

Respectfully submitted,

By /Leslie A. Van Leeuwen Reg. No. 42,196/
Leslie A. Van Leeuwen, Reg. No. 42,196
Van Leeuwen & Van Leeuwen
Attorneys for Applicant
Telephone: (512) 301-6738
Facsimile: (512) 301-6742